Homework 7

In this homework assignment, you will learn the basics of writing functions. Further, you will learn a little about random numbers.

Average, Variance, and Standard Deviation

For this homework assignment you will write a few basic functions to compute the average, variance, and standard deviation of a set of numbers. A sample execution of the program is provided below:

```
Enter seed: 100
Generating 10 random numbers from 1 to 6...
The 10 numbers are: 2 4 4 2 6 4 6 3 4 5
The average is: 4.0
The std. deviation is: 1.3416407864998738
The variance is: 1.8
```

I've provided the function `throw_dice()`, which will generate a random number from 1 to 6 (a standard die has six faces):

```
# Generates a single random value between 1 and 6
def throw_dice():
     return random.randint(1, 6)
```

The `throw_dice()` function will generate a single random value (an integer), each time it is called. I've also provided you with a `main()` function that uses `throw_dice()` to create a list of random numbers:

```
# Main program. We'll be writing our program inside of a function
# this time. Write the code to test your other functions
#(compute_average, etc) inside of this function.
def main():
     seed = int(input("Enter seed: "))
     random.seed(seed)
     print("Generating 10 random numbers from 1 to 6...")
     random_numbers = [throw_dice() for x in range(0, 10)]

     # Your code calling/testing the functions here
```

All of this code is provided in the `hw7.py` file on the course website. Now on to the code you have to write. You'll write functions to compute the average, variance, and standard deviation of the sequence of random numbers we just generated. **You must write the code for these operations inside of the provided function skeletons, otherwise your program will not run correctly on Web-CAT. You may not use any built-in functions (except sqrt() and pow()) to compute these values, and you must iterate through the lists yourself.** The TAs will retroactively deduct 50% if you don't follow these rules.

Inside of the provided file should be a function called `compute_average()`.This function should take a list of integers as the parameter and return the average:

```
# returns the average of the numbers in list "numbers"
def compute_average(numbers):
     # get rid of the pass statement and implement me
     pass
```

Test your `compute_average()` function from `main()`. To be sure, you may need to write out the average on paper or find an online calculator.

Next, write the function `compute_variance()`. **You must call** the `compute_average()` function inside of the `compute_variance()` to receive full credit. The variance should be computed as:

$$V = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2$$

Where N is the length of `numbers`, x_i is an element of `numbers`, and μ is the average of all the elements in `numbers`.

```
# computes the variance of the numbers in list "numbers"
# must use compute_average()!
def compute_variance(numbers):
     # get rid of the pass statement and implement me
     pass
```

Finally, write the function `compute_stddev()` which takes a list of numbers, just like `compute_variance()`and then returns the **population** standard deviation of the random sequence. **You must use** the `compute_variance()` function to calculate the standard deviation. The standard deviation should be computed as `sqrt(variance)`.

```
# computes the standard deviation of the elements in list "numbers"
# must use compute_variance()!
def compute_stddev(numbers:)
     # get rid of the pass statement and implement me
     pass
```

Hint: You will need to `import math` to use functions like `pow()` and `sqrt()`.

Once you feel confident that your code is working the way it should, then go ahead and submit it to Web-CAT. Make sure the function names, and number of parameters, are the same as the description. The TAs will check that you have followed these instructions and mark you down if haven't. You will only receive half credit for a function if it works but you haven't followed directions.

Mini-intro to Randomness
(Originally written by Dwight Barnette for CS1124)

Random numbers in programming are almost always pseudo-random numbers. They are generated from a mathematical formula and thus are not truly stochastic. If the formula is not "seeded" differently for each execution **then the same sequence of numbers will be generated each time**.

In Python the random capability is provided by the `random` module. It must be imported before use with the import statement. Here are some of the useful functions provided by the random module.

`random.seed(intValue)` - Initializes the random sequence starting point using the `intValue`.
`random.random()` - Returns the next real/float value in the range: [0.0, 1.0).
`random.randint(a, b)` - Returns a random integer in the range: [a, b].

The above list is only a sampling of the Python random methods. Other methods are available. Interested readers should refer to the online Python language documentation:

http://docs.python.org/3/library/random.html

What to Submit

For this assignment you should submit your **hw7.py** file.

This assignment will be graded automatically. Test your programs thoroughly before submitting them. Make sure that your programs produce correct results for every logically valid test case you can think of. Do not waste submissions on untested code, or on code that does not run with the supplied code from the course website.

Web-CAT will assign a score based on runtime testing of your submission; your best score will be counted; the TAs will later verify that your best submission meets the stated restrictions, and assess penalties if not.

To submit this assignment:

1. Visit http://web-cat.cs.vt.edu in your web browser.
2. Enter your Virginia Tech PID and password in the appropriate fields on the log-in screen, and make sure that **Virginia Tech** is selected as the institution. Click **Login**.
3. The Web-CAT home screen will display useful announcements and assignments that are currently accepting submissions. Find the assignment that you want to submit in the table, and click the "Submit" button next to it.
4. Click the **Browse...** button and select the file you want to upload. The homework assignments and programming projects for this course should be self-contained in a single **.py** file, so you can simply select that one file.
5. Click the **Upload Submission** button. The next page will ask you to review your selection to ensure that you have chosen the right file. If everything looks correct, click **Confirm**.

The next page will show that your assignment is currently queued for grading, with an estimated wait time. This page will refresh itself automatically, and when grading is complete you will be taken to a page with your results.

Pledge

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the submitted file:

```
#  <include a description of the purpose of this file/project/package>
#
#  @author <name and surname> (your VT PID)
#  @date   <the date>
#
# Virginia Tech Honor Code Pledge
#  On my honor:
#
#  - I have not discussed the Python language code in my program with
#    anyone other than my instructor or the teaching assistants
#    assigned to this course.
#  - I have not used Python language code obtained from another student,
#    or any other unauthorized source, either modified or unmodified.
# - If any Python language code or documentation used in my program
#    was obtained from another source, such as a text book of coarse
#    notes, that has been clearly noted with a proper citation in
#    the comments of my program.
#  - I have not designed this program in such a way as to defeat or
#    interfere with the normal operation of the Web-Cat Server.
#
#  <your name>
```

Failure to include this pledge in a submission will result in the submission being disallowed during code review.